ISSN 2224-087X. Електроніка та інформаційні технології. 2020. Випуск 14. С. 49–54
Electronics and information technologies. 2020. Issue 14. P. 49–54

УДК 004.032.26; 004.855.5 DOI: https://doi.org/10.30970/eli.14.4

ВИКОРИСТАННЯ БІБЛІОТЕКИ ПРОГРАМ TENSORFLOW ДЛЯ РЕАЛІЗАЦІЇ НЕЙРОМЕРЕЖ НА МІКРОКОНТРОЛЕРАХ ФІРМИ INFINEON TECHNOLOGIES AG

А. Цемко [1], Я. Берко[1], В. Бігдай[2], З. Любунь[1]

[1]Львівський національний університет імені Івана Франка,
вул. Ген. Тарнавського, 107, 79017 Львів, Україна

[2] Infineon Technologies AG,
вул. Луганська, 20, 79000 Львів, Україна

Останнім часом суттєво збільшується кількість задач розв'язання яких через свою складність стає можливою тільки з використанням методів машинного навчання. Виходячи з означення машинного навчання, можна зрозуміти, що його використовують у випадках, коли традиційним способом задача погано формалізується і визначення правил чи закономірностей для неї утруднене. Так, наприклад, використовуючи традиційні парадигми програмування, написання алгоритму розпізнавання зображень в більшості випадків не дає задовільних результатів. В такому випадку звертаються до машинного навчання, яке визначає правила і закономірності, аналізуючи вхідні і вихідні дані, і надає можливість з певною імовірністю провести розпізнавання зображення.

Головною проблемою використання нейронних мереж на сьогодні є те, що їх розмір, а через це і кількість математичних операцій, займає дуже великий обсяг. Це призводить до ускладнення і зниження швидкодії алгоритмів, які використовують нейронні мереж. Зазвичай, задачі з використанням нейронних мереж потребують отримання швидкого результату, і тоді, для використання нейронних мереж на IoT пристроях підключають віддалений сервер, який отримує набір даних, пропускає їх через нейронну мережу, і надсилає дані у відповідь. У такого способу організації використання нейронної мережі є очевидні недоліки, такі як: потреба у постійному і швидкому інтернет підключенні (що іноді не є можливим); потрапляння конфіденційної інформації користувача у інтернет мережу (що підвищує ризик перехоплення цих даних зловмисниками); потужний віддалений сервер, який спроможний паралельно обраховувати достатню кількість запитів від пристроїв користувачів, задля надання мінімального часу затримки між отриманням і надсиланням запитів і т.д.

Альтернативним способом використання та організації нейронних мереж на IoT пристроях є їх квантування (перетворення значень ваг нейронної мережі в цілочисельні значення з значень із плаваючою комою), що значно зменшує розмір навченої нейронної мережі, і дозволяє, в свою чергу, обчислювати математичні операції нейронних мереж на процесорах, призначених для роботи з цілочисельними типами даних. Такий спосіб організації нейронних мереж не потребує доступу до інтернету, що, в свою чергу, виключає час очікування на передачу і приймання даних по бездротових мережах, а також виключає можливість перехоплення конфіденційних даних користувача під час передавання або приймання даних.

Ключові слова: штучні нейронні мережі, нейронні мережі на мікроконтролері, квантування нейронних мереж.

Створення нейронних мереж і оптимізація процесу навчання, за якого відштовхуючись від значної кількості навчаючих пар змінюються значення матриць ваг, є складним і довгим процесом. Також, у процесі створення нейронних мереж ключовим є підбирання структури нейронної мережі спираючись на поставлену задачу. Процес змінення структури мережі: додавання нових шарів, змінення кількості нейронів, застосування інших типів мереж; займатиме дуже багато часу (не враховуючи, процесу навчання і валідації навченої мережі). Через це, для створення нейронних мереж використовують бібліотеки та фреймворки, які дозволяють швидко створювати, перебудовувати та зручно досліджувати нейронні мережі.

Однією із популярних і доступних бібліотек для швидкого створення нейронних мереж є бібліотека TensorFlow [1] мови програмування Python від Google, з відкритим кодом. Бібліотека TensorFlow, в порівнянні з іншими вільно поширюваними бібілотеками, - єдина бібліотека, яка дозволяє згортати навчену мережу, тим самим зменшуючи її об‘єм, і надаючи можливість використовувати її на мікроконтролерах з дуже малим обсягом пам‘яті.

TensorFlow дозволяє створювати нейромережі за допомогою низькорівневої API, або високорівненої API - Keras. Для вивчення та ознайомлення з машинним навчанням, ми рекомендуємо спершу використовувати високорівневу API Keras, бо вона надає більш просте розуміння структури нейромереж.

Також, для зручності спостереження процесу навчання мережі, існує бібліотека TensorBoard (встановлюється разом з TensorFlow), яка надає можливість контролювати процес навчання мережі у реальному часі з оновленням у часі кожні 30 секунд. Окрім того, за допомогою TensorBoard, можна переглядати граф мережі, перевіряти зміни у графі та виводити додаткові важливі дані для навчання.

Для дослідження можливостей бібліотеки TensorFlow, була взята мережа, що апроксимує функцію sin(x). Дані для навчання складались з 360 точок, отриманих шляхом табуляції функції sin(x) в межах від 0 до 2π.

Структура мережі, яку ми обрали для цієї задачі, складається з п’яти повнозв’язних шарів із загальною кількістю нейронів – 97. Вигляд графу мережі що відповідає структурі мережі в TensorBoard показана на рис. 1.

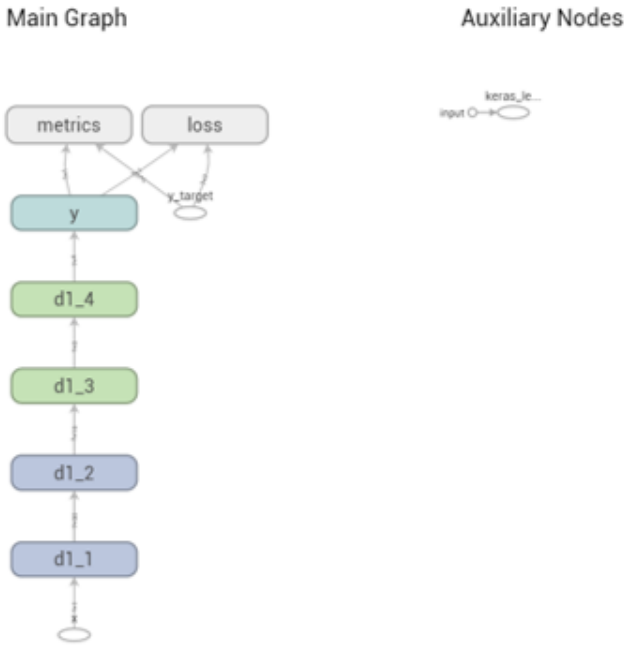

Рис. 1. Вигляд графу п'ятишарової повнозв'язної мережі у TensorBoard

Після навчання мережі протягом 1000 епох (епоха – це повне проходження по всіх навчаючих парах), ми отримали навчену нейронну мережу котра апроксимує задану функцію з середньоквадратичною похибкою, котра не перевищує значення 8.6e-05. Результати апроксимації показані на рис. 2.

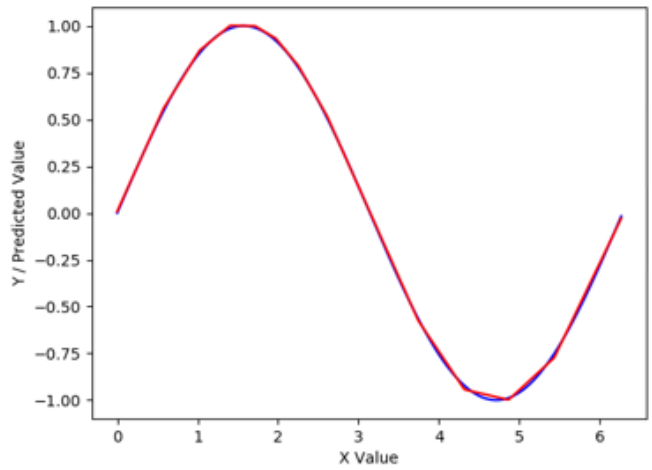

Рис. 2. Резльтута апроксимації нейронною мережею: блакитна крива – вхідні дані, червона крива – результат апроксимації нейронною мережею.

Для імплементації навченої нейронної мережі на мікроконтролері необіхдно суттєво зменшити об'єм необхідної пам'яті для збереження ваг навченої мережі. Це можна зробити шляхом переходу до цілочисельних значень ваг мереж [2,3]. Таке перетворення називається квантуванням нейронної мережі. В нашому випадку, ми перетворили ваги нейронної мережі до типу даних int8.

Бібліотека TensorFlow починаючи з версії 2 дозволяє реалізовувати квантування нейронної мережі після її навчання (в майбтуьному, планується реалізації функції для квантування значень ваг мережі під час її навчання). На сьогодні, TensorFlow дозволяє провести квантування ваг мережі у типи значень int8, int16 та float16. Слід зауважити що, інтерпретатори TensorFlow Lite сприймають лише знакові цілочисельні типи, і при викилку мережі з беззнаковими типами (uint8, uint16) буде отримано овідомлення про помилку і несумісність типів. Також, квантування нейронної мережі може бути двох видів: повне та часткове. Повне квантування включає в себе зміну як ваг нейронної мережі так і вхідних і вихідних сигналів. Часткове квантування дозволяє змінювати лише ваги нейронної мережі, залишаючи вхід і вихід нейронної мережі у початковому типі даних. Оскільки, обраний нами мікроконтролер працює лише з цілочисельними значеннями, то нами було проведено повне квантування нейронної мережі до цілочисельного типу int8.

Після конвертування нейронної мережі, отримується файл розширення .tflite, який зберігає в собі структуру мережі. Для ознайомлення з виглядом моделі, можна використати додаток Netron, який дає змогу отримати структуру мережі, як це зображена на рис. 3.

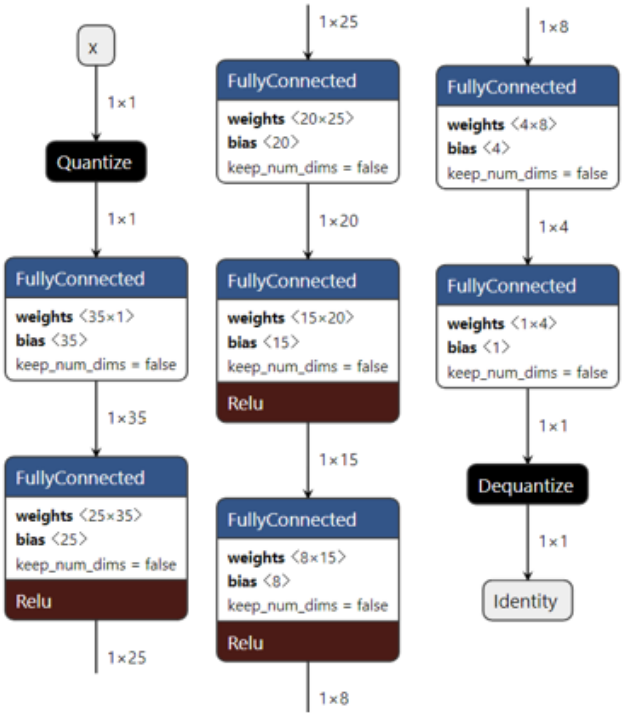

Рис. 3 Структура конвертованої нейронної мережі у додатку Netron

Для реалізації нейронної мережі на мікроконтролері отриманий конвертований .tflite файл, необхідно перевести у HEX формат і потім використати його у проекті на мікроконтролері.

Нами було використано мікроконтролер фірми Cypress Semiconductor, а саме - CY8CKIT-062-WiFi-BT [4]. Бібліотека TensorFlow надає велику кількість прикладів, для використання мережі на різних мікроконтролерах, тому базуючись на одному з таких прикладів, ми реалізували проект на мікроконтролер з нашею навченою нейронною мережею. Налаштували вивід через UART і додавши логіку виклику нейронної мережі. Ми отримали значення нашої апроксимуючої функції показані на Рис. 4, що свідчить про те, що квантована нейронн мережа працює корректно на нашому мікроконтролері.

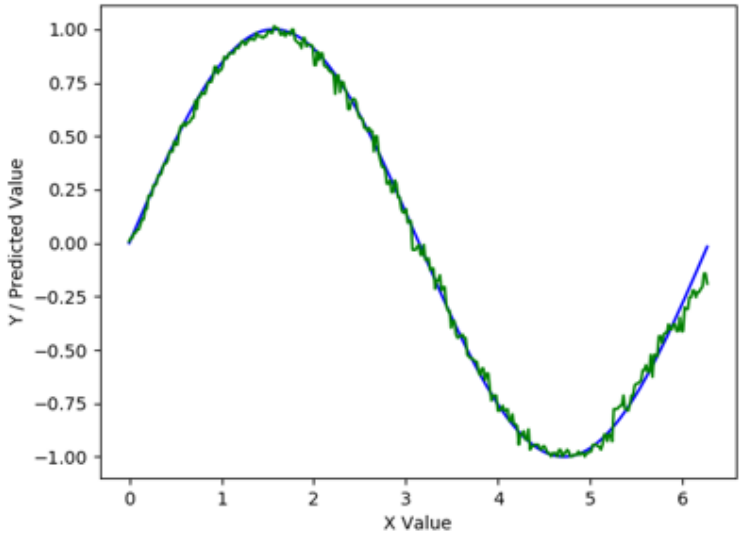

Рис. 4. Результати роботи квантованої нейронної мережі
на мікроконтролері CY8CKIT-062-WiFi-BT:
синя крива – вхідні дані для навчання,
зелена крива – апроксимація квантованою мережею

Проаналізувавши отримані результати можна побачити, що квантована нейронна мережа на мікроконтролері працює корректно. З цього можна зробити висновок, що складніші задачі з застосування нейронних мереж можливо реалізувати на мікроконтролері, враховуючи максимальний можливий розмір нейронної мережі відносно обсягу пам’яті на мікроконтролері

Список використаної літератури

1. TensorFlow Guide [Електронний ресурс] – Режим доступу: https://www.tensorflow.org/guide - 30.09.2020
2. Post-training quantization [Електронний ресурс] – Режим доступу: https://www.tensorflow.org/lite/performance/post_training_quantization - 11.11.2020

3. *Karpin O.* Method of Neural Network Training with Integer Weights / O. Karpin, V. Mandziy, Z. Liubun, V. Rabyk // Proceedings of the XI[th] International Scientific and Practical Conference "Electronics and Information Technologies" (ELIT - 2019), September 16 – 18, 2019, Lviv, Ukraine. P. 168 – 172. doi: 10.1109/ELIT.2019.8893349 .
4. CY8CKIT-062-WiFi-BT PSoC® 6 Wi-Fi Pioneer Kit Guide [Електронний ресурс] /
5. Документація Cypress Semiconductor щодо мікроконтролера PSoC 6 6 Wi-Fi Pioneer Kit – Режим доступу: https://www.cypress.com/file/407731/download – 30.09.2020 р.

USING TENSORFLOW ECOSYSTEM OF TOOLS FOR IMPLEMENTING NEURAL NETWORKS ON CYPRESS SEMICONDUCTOR MICROCONTROLERS

A. Tsemko[1], B. Yaroslav[1], V. Bihday[2], Z. Liubun[1]

[1]Ivan Franko National University of Lviv,
107 Tarnavsky St., UA–79017 Lviv, Ukraine

[2] Infineon Technologies AG,
20 Luhanska St., UA–79000 Lviv, Ukraine

Today, a lot of difficult tasks require the implementation of neural network methods. According to the definition, neural networks are used to recognize the correlation between input and output data, that cannot be easily formalized by a programmer. For example, the programmer cannot formalize basic relationships between image pixels and recognizable items on the picture. In this case, the traditional programming paradigm is replaced by machine learning.

The problem of using neural networks on difficult tasks is a large amount of data. Also, the process of using neural networks requires a lot of floating-point operations. Usually, a neural network task requires a real-time response, and for this reason, neural networks use cloud services. Cloud services get input data from an IoT peripheral, process it, and send the response of output data back to an IoT peripheral. Such a way of neural network usage organization with IoT peripherals has some disadvantages. For instance, an IoT peripheral should have a connection to a network, personal user data can be stolen due to transporting, cloud service should be powerful and has a multithreading module for real-time response to a large number of the devices at the same time.

An alternative method of using neural networks on IoT peripherals is quantization. Quantization is a technique that reduces the size of neural network weight by converting it to a smaller data type. Using integer quantized values boosts the speed of calculation and reduces the size of the whole neural network. This method avoids a connection to a powerful cloud service, so personal data do not leave an IoT peripheral and cannot be stolen.

Key words: neural networks, quantization, machine learning, neural networks on IoT peripherals and microcontrollers.

Стаття: надійшла до редакції 05.11.2020,
доопрацьована 10.11.2020,
прийнята до друку 10.11.2020